LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

1

LA-UR -89-760

CONF-890693 /

Received by OSTI

APR 0 6 1989

LA-UR--89-760

DE89 009407

TITLE THE DYNAMICAL BEHAVIOR OF CLASSIFIER SYSTEMS

AUTHOR(S) Stephanie Forrest

SUBMITTED TO Third International Conference on Genetic Algorithms, Washington, DC, June 4-7, 1989

Comments Welcome Version 1.0

The Dynamical Behavior of Classifier Systems[*]

Stephanie Forrest, Center for Nonlinear Studies, Los Alamos National Laboratory[†]

John H. Miller, Santa Fe Institute and Carnegie-Mellon University[‡]

30 January 1989
Revised: 10 February 1989

[*]Acknowledgments: The financial support of the Santa Fe Institute and Los Alamos National Laboratory are gratefully acknowledged. Computational support was provided by the Knowledge Systems Laboratory at Los Alamos. Rick Riolo was kind enough to supply us with some sets of classifiers. The paper also benefited from early discussions with John Holland, and extended discussions with Stuart Kauffman.

[†] CNLS, MS-B258, Los Alamos National Laboratory, Los Alamos, NM 87545 (Sxf@lanl.gov).

[‡] Santa Fe Institute, 1120 Canyon Road, Santa Fe, NM 87501 (Miller@sfi.santafe.edu).

1. Introduction

Classifier systems are quite complicated, in terms of both their components and behavior. This complexity is understandable given the wide spectrum of activity they are intended to model. Unfortunately, the complexity of these systems also makes it difficult to understand them analytically. Previous analysis has focused on specific components of the classifier system, for example, the genetic algorithm or the bucket brigade. The lack of a unified theory has led users of these systems to rely on *ad hoc* methods for choosing representations and parameter settings. Recent results (Riolo, 1988) indicate that classifier systems can be very sensitive to particular encodings and parameter choices. In this paper, we propose a methodology for studying the interactions among various components of the classifier system architecture.

Classifier system behavior can be studied by analyzing an equivalent dynamical system. The dynamical systems perspective has important consequences for classifier systems: fundamental properties emerge which may enhance or destroy the abilities of such systems. Accompanying the purely descriptive elements of the analysis, are important prescriptive ideas. Different aspects of classifier systems can be directly linked to these emergent dynamical properties with implications for the effective design of classifier systems. A variety of questions can be addressed using the methodology developed here, including, but not limited to: How many classifiers are required in the system before interesting behavior can occur? What is the likelihood that chains of classifiers will form? How dense will these chains be? What is the impact of specificity on these properties? What is the impact of learning and representation on the dynamical behavior? How stable are these systems?

By viewing a set of classifiers as a network, it is possible to investigate the topological properties of various collections of classifiers. The connection between classifiers and networks is well known (Forrest, 1985). Section 2 defines a mapping between classifier systems and a class of finite automata called random Boolean networks (Kauffman, 1968). The dynamical properties of random Boolean networks have been studied extensively (Kauffman, 1984) (Derrida and Stauffer, 1986) (Derrida and Weisbuch, 1986), and general techniques have been developed for determining these properties. This work shows that dynamical behavior can be characterized by a set of emergent properties. In this paper, we link these properties to important aspects of classifier system behavior.

The paper first develops the mapping between classifier systems and Boolean networks, and discusses various emergent properties. Section 3 presents preliminary numerical results, and Section 4 discusses the implications of this approach.

2. Classifiers as Dynamical Systems

The analysis relies on three premises: (1) a mapping between classifier systems and Boolean networks can be defined that preserves the relevant dynamical properties of classifier systems, (2) the dynamic behavior of Boolean networks is dominated by a set of emergent properties inherent in the network structure, and (3) these emergent properties have important implications for the behavior of classifier systems. These three premises are discussed in order.

2.1 Random Boolean Networks

A Boolean network consists of a set of nodes, each of which has two possible states, 0 or 1. The state of any given node at time $t+1$ is determined by a subset of the states of the other nodes in the network (those with directed arcs into the given node) at time t. A predetermined (and time invariant) Boolean function is associated with each node. The variables of the function correspond to the states of the connected nodes. The Boolean functions can vary for different nodes, as can the number and location of the input nodes. A three-node Boolean network is shown in Figure 1.

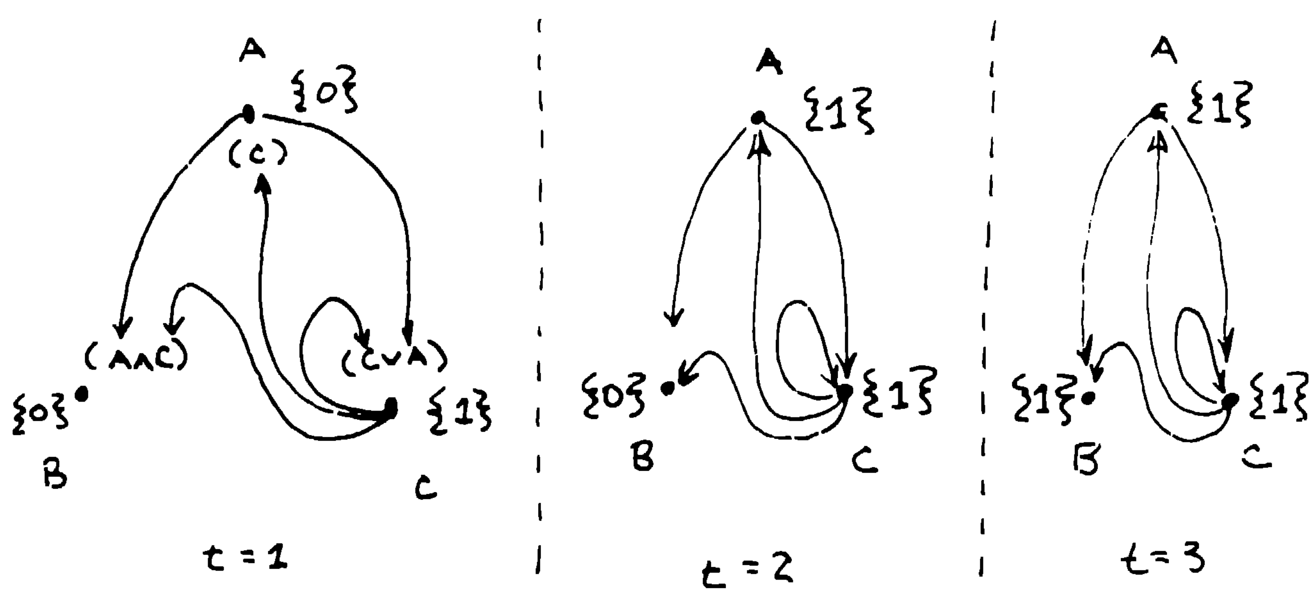

Figure 1. An Example Boolean Network

A random Boolean network (RBN) is a Boolean network in which the connectivity pattern and the Boolean functions are assigned stochastically. For example, an RBN could be formed for an n node network, where every node is randomly connected to two other nodes by either an AND or an OR function, also selected randomly. A specification, such as the one just mentioned, defines one class of RBNs—each individual member of a particular class will have similar emergent properties. An important question is whether classifier systems correspond to a specific class of RBNs.

2.2 A Mapping from Classifier Systems to Boolean Networks

The mapping from a classifier system (CS) to a Boolean network (BN) is defined in stages. Initially, CS is a simple two-condition classifier system with negation, without pass-through, and with no provisions for limited size message lists or bidding. Negation is only allowed on the second condition. The mapping is defined as follows:

1. Assign one node in BN for every possible message that could be posted by CS or its environment. Each node will be in State 1 exactly when the corresponding message would be posted to the message list in CS, and it will be in State 0 otherwise. The Boolean function associated with each node enforces this behavior.

2. For each classifier $c_i \in CS$ (c_i is of the form *Condition-1*$_i$, *Condition-2*$_i$: *Action*$_i$), construct the Boolean function f_i for the node corresponding to *Action*$_i$ such that f_i is

true whenever c_i would fire. Specifically f_i will be of the form $((m1_1 \vee m1_2 \vee \ldots) \wedge (m2_1 \vee m2_2 \vee \ldots))$ where the $m1_j$ and and $m2_k$ are the sets of messages that match Condition-1 and Condition-2 respectively. f_i will be of the form $((m_1 \vee m_2 \vee \ldots) \wedge \neg(m2_1 \vee m2_2 \vee \ldots))$ if the second condition is negated.

3. The node corresponding to $Action_i$ is assigned f_i as its Boolean function if there is no other Boolean function defined on that node; otherwise, f_i is combined disjunctively with the existing Boolean function. This latter situation arises when two or more classifiers have identical action messages.

It is straightforward to show that this mapping preserves the functional behavior of a simple classifier system. That is, for each possible state of the message list at time t (corresponding to a set of nodes in the Boolean network that are in State 1), the message list produced by the classifier system at time $t+1$ will be equivalent to the set of nodes in State 1 at time $t+1$.

To extend the mapping for pass-through, we assume that the pass-through operation is defined on the first condition. A classifier with pass-through is effectively distributed across its various output nodes. Specifically,

4. For each c_i, construct the set A_i of all c_i's possible action messages, and for each $a_j \in A_i$ construct the set $M1_{ij}$ of possible activating messages for that action message. Each message in $M1_{ij}$ must satisfy both the constraints (*i.e.* match) of Condition-1 of c_i and the output message a_j. The set $M2_i$ is constructed by taking all messages that could possibly match Condition-2. The result is a set of $< a_j, M1_{ij}, M2_i >$ triples for each classifier c_i.

5. For each $< a_j, M1_{ij}, M2_i >$ triple, the Boolean function at node a_j is augmented as in Step 3 with a clause that specifies the conditions under which the output message a_j would be produced.

Figure 2 shows the mapping for three classifiers. The three-bit two-condition classifier 1#0, #01; 000 corresponds to the Boolean function $((100 \vee 110) \wedge (001 \vee 101))$ on the 000 node. The classifier 00#, ¬111; 11# is distributed across the nodes 110 and 111 which correspond to the possible passed-through messages that it could produce. This classifier also illustrates the use of negated conditions. The classifier 001, 110 ; 110 shares an output node with the second classifier; the Boolean expressions for each classifier are combined disjunctively.

By restricting the number of nodes that can be in the "1" state at any instant, message lists of various restricted sizes can be simulated. In the case where there are more nodes whose Boolean functions evaluate to 1 than there are slots on the message list, a nondeterministic procedure is used to select which nodes are actually set to State 1. In the case where there is a strength associated with each classifier, that strength can be apportioned among the various Boolean functions that implement the classifier. The Boolean function associated with each node then becomes a probablistic function of the strengths associated with the various clauses.

The Boolean networks that correspond to classifier systems have important structural characteristics. There are two types of nodes in these systems: internal and external. External nodes have empty Boolean functions, and correspond to messages that might be generated by the input interface of the classifier system. Internal nodes have nonempty Boolean functions and correspond to messages that can be sent by a given classifier. In Figure 2, Nodes 000, 110, and 111 are internal and the remaining nodes are external.

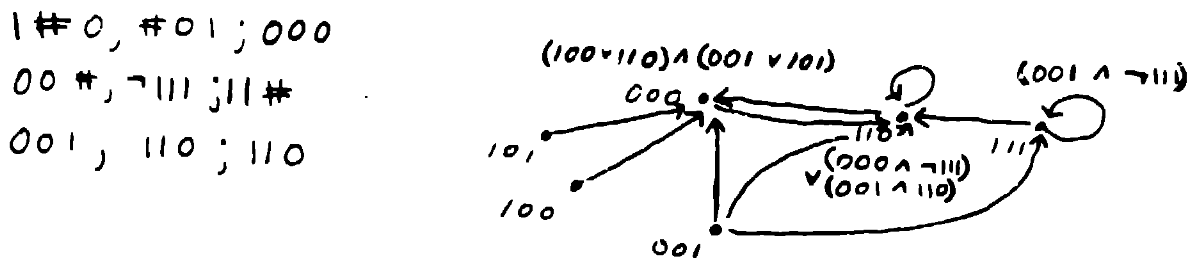

Figure 2. An Example Mapping

The Boolean functions defined on internal nodes have a regular structure; this can be an important determinant of global behavior for certain RBNs

Once a classifier system has been mapped into an equivalent Boolean network, its dynamic behavior can be studied. The state of each node corresponds to whether or not its message is currently posted on the classifier systems message list. By setting the nodes in the Boolean network to correspond with the initial set of messages present in the system, temporal behavior of a classifier system can be observed by iterating the network. Messages from the input interface are mapped to the external nodes of the network. Since external nodes have empty Boolean functions, none of them will fire after the initial time step. For some questions, such as determining the amount of internal connectivity in the network, external messages are not particularly relevant. For other questions, input messages from the environment can be simulated by periodically firing some of the external nodes during the iterations.

The configuration[1] of a particular classifier system has a direct impact on its Boolean network structure. The number of internal nodes in the network is related to the number of classifiers in the system and to the use of pass-through. An internal node exists for every unique message sent by a classifier. Thus, larger numbers of classifiers or the use of pass-through will in general imply more internal nodes. However, as the number of classifiers increases, the likelihood of duplicate messages increases, implying that the number of internal nodes will not increase exactly linearly with size.[2] The particular Boolean function

[1] Where configuration includes the number of classifiers, negation, pass-through, the proportion of #'s, the type of learning, etc.

[2] In an 8-bit system there are 256 possible messages, and therefore even modest numbers of classifiers, say, 60, may begin to saturate the system.

and the number of input arcs on each internal node are related to the number of input conditions from various classifiers and the proportion of # symbols. As the number of #s increases in a condition without pass-through, the connectivity of the corresponding node should increase exponentially. Pass-through complicates the calculation, however, because the number of output nodes is increasing at the same time (but not necessarily the same rate) as the number of input messages. Different learning algorithms may also affect network structure. For example, some algorithms may encourage convergence around some sets of important messages. The details of different classifier system configurations and their network properties are explored in Section 3.

A particular classifier system can be mapped to an equivalent Boolean network. Because different configurations of (randomly generated) classifier systems have distinct structural properties, we can study the dynamics of any particular configuration by generating RBNs with the structural properties (mean number of input arcs, proportion of internal nodes, *etc.*) corresponding to that configuration. It may be possible to use a similar approach for classifier systems after learning. This will require a careful study of the differences between classifier systems before and after learning. Of particular consequence is how the overall topological structure (in the mapped representation) changes after learning. For example, some learning algorithms might take advantage of existing topological structures (cycles, *etc.*) and build associations between the Input/Output interface and the existing structures. Others might construct new topological structures that are significantly different from those that exist in randomly generated classifier systems. As shown above, different configurations will imply different underlying network structures. By finding the emergent properties of these different structures, a link between system design and performance can be derived and exploited.

2.3 Emergent Properties of Random Boolean Networks

The emergent properties of RBNs depend on the number of nodes, the number of connections between each node, and the Boolean functions employed (Kauffman, 1984). Once these characteristics are known, the typical behavior of different classes of such networks can be firmly established. The configuration of a classifier system will have a direct effect on the aforementioned network properties. Thus, an understanding of the dynamics of Boolean networks vis-à-vis these properties, provides insights into the dynamics of various types of classifier systems.

A given Boolean network with n nodes has 2^n possible states ($\{0,1\}^n$). The Boolean functions and connections among nodes imply a deterministic state transition function. Given that the network is deterministic and finite, it must eventually fall into some state cycle. Different initial conditions may, however, cause the network to enter different state cycles. All points in the state space are either part of some state cycle, or they lie on a trajectory that leads to a cycle.

Although this is the first analysis of Boolean networks that correspond directly to classifier systems, networks with similar structures have been analyzed (Kauffman, 1984). The results of these analyses (Kauffman, 1984, pp. 151-2) indicate a variety of important dynamical network properties. For example, the actual number of distinct state cycles is on the order of the square root of the number of nodes. For a network with 10,000 nodes, this would imply only 100 distinct state cycles. Cycle lengths are also typically small, with median cycle length again on the order of $n^{0.5}$ where the theoretical maximum is 2^n (in the 10,000 node case this is the difference between 100 and $2^{10,000}$). A large fraction of the nodes (60–80%) tend to fix to either always on or always off in a given state cycle. Different cycles tend to have similar states, with hamming distances[3] between 1–10%. State cycles tend to be stable to most one node perturbations, *i.e.*, if one node in the network randomly changes state, the network usually does not enter a new state cycle. Many of these properties have direct implications for classifier systems. Moreover, some unexplored properties of these networks are also relevant to classifier systems, for example: the impact of randomly firing external nodes, the propagation paths (the Boolean net analog of an execution trace) caused by the activation of particular subsets of nodes, the use of probabilistic Boolean functions and restricted size message lists, recording basins of attraction for different sets of external nodes (for example, if any combination of external messages led to the same basin of attraction, it would indicate that the classifier system wasn't differentiating well between various inputs), measuring the lengths of transients (transients are likely to be quite important for classifier systems with frequent input from the environment), and state cycles for sub-networks (global states are likely to be less important than states for functional sub-pieces of a network).

2.4 The Importance of Emergent Properties to Classifier Systems

The preceding discussion suggests that the dynamic behavior of classifier systems are dominated by a set of emergent properties. The existence of these properties may impose major constraints on the performance of any classifier system. Therefore, an understanding of the link between some of the emergent properties and classifier behavior is important.

The classifier system's architecture derives much of its power through the formation of chains of rules. Such chains support internal reasoning processes that allow a classifier system to exhibit more than stimulus/response behavior. The sub-state cycles (cycles in sub-networks) that emerge in Boolean networks are closely related to chains in a classifier system. Any cyclic chain of classifiers will correspond to a cycle of states in the Boolean network, and any cycle of activity in a group of nodes will correspond to some set of classifiers activating one another. Based on this connection, we predict that the likelihood of chain formation in classifier systems is closely tied to the system's configuration. Critical values probably exist that catalyze the formation and survival of chains. Section 2.3 indicates that the number of classifier chains that form is relatively small and that they tend to have few members. The ability of classifier systems to exhibit self-sustaining activity, that is, to generate internal activity in the absence of external input, is also likely tied to certain configuration parameters. At some level of self-sustaining activity, systems should be able to operate with little or no environmental input and be able to form large internal representations. Too much internal activity is expected to hurt performance, since the environment is a crucial component of learning in classifier systems.

If the Boolean networks corresponding to classifier systems have similar properties as those described in the literature, most classifier chains will be very similar to one another in terms of the actual messages that are posted. Once a chain is entered, it should be stable to small perturbations. If a perturbation does invoke a new chain, typically only a small subset of the existing chains are likely to be available. That there is a tradeoff

[3] Two global states of a Boolean network can be compared by assigning each node in the network one bit position in a binary string and setting the bit position according to the current state of the corresponding node. The hamming distance between two such strings then provides a measure of similarity between the two states.

between system stability and ultimate performance. Stable systems will be able to operate gracefully in the presence of unusual external stimuli, and under the influence of changing structure due to learning. Nonetheless, the difficulty of implementing new chains may be problematic in excessively stable systems.

Finally, the potential for most of the messages (*i.e.*, the nodes corresponding to messages) to become fixed (either on or off) exists – a phenomenon called "freezing." By considering the truth table that defines a particular Boolean function, it is possible to determine which combinations of input values produce 1's and which ones produce 0's. Thus, for a binary disjunction, there are 75% 1's and 25% 0's in its truth table, and given random inputs the result will be a 1 3/4 of the time and a 0 1/4 of the time. For Boolean networks, an important statistic is this ratio of 1's to 0's in the functions defined on the nodes. Boolean functions that have either an abnormally high or low percentage of ones in their truth table, tend to freeze *ceteris paribus*. Frozen messages may hinder attempts at forming a system that can actively respond to new situations. Moreover, algorithms that reward or penalize classifiers based only on the presence or absence of the associated message at a given time, may be adversely affected by frozen messages. A close link between the form of classifier conditions and the percentage of ones is developed in the next sections.

The above elements combine into a potentially powerful descriptive and prescriptive methodology. Many of the important performance characteristics of a classifier system are likely to be related to its dynamical behavior. Through the use of the above mapping, connections between system design and dynamical behavior can now be derived. This connection can be used both to understand current system performance and in the design of new systems with improved performance characteristics.

3. Results

The emergent properties of a Boolean network depend on a relatively small set of defining characteristics. The major characteristics which determine a network's behavior are the number of nodes and their associated Boolean functions. A complete analysis requires both an understanding of how different parameter choices in standard classifier systems influence these properties, as well as how these properties affect the dynamic behavior of the network. The results reported here are preliminary, but they focus on both of these issues.

All experiments were conducted using an 8-bit classifier system. An 8-bit system, rather than more common 16- or 32-bit systems, was used in order to simplify computation. From the Boolean network perspective, an 8-bit classifier system with 20 classifiers is roughly equivalent to a 16-bit classifier system with over 1000 classifiers. We expect that relatively simple scaling relations exist that will allow the results to be applied to arbitrarily-sized systems. Each reported data point is a mean computed from 30 randomly generated classifier systems.

3.1 Internal and External Nodes

The distinction made in Section 2.2 between external and internal nodes separates messages produced by the environment from internal message-passing. For each network there are 256 possible nodes (messages), but not all of them are part of every Boolean network. If there were a message, for example, that the environment never posts and no classifier can respond to, then the node corresponding to that message would not be included in the network. Thus, there are two values of interest: the number of actual internal nodes and the number of actual external nodes. For the structural properties studied in the following sections, the ratio of internal to external nodes is the of interest, and it is shown in Figure 3. The ratio should remain constant as the number of classifiers increases (since each new classifier creates new internal and external nodes at the same rate), until the system begins to saturate. At saturation, the addition of a new classifier will not always produce a constant number of new nodes, since the nodes used by the classifier may already be in the network. The figure shows that the ratio increases gradually, indicating that external message nodes saturate more quickly than internal nodes.

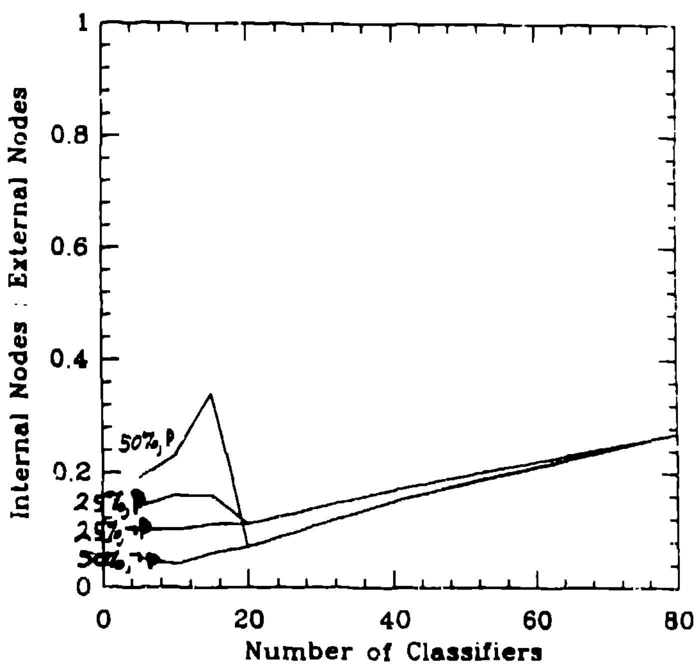

Figure 3. Internal and External Nodes

3.2 Connectivity

Previous results (Kauffman, 1984) suggest that the average number of input arcs per node is an important determinant of a network's dynamical properties. Figures 4 and 5 show the average number of input arcs per internal node. The first figure shows the average for both internal and external arcs, while the second one considers only those arcs coming from other internal nodes. As expected, the use of more #s in the conditions significantly increases the amount of connectivity. The average number of arcs also increases with the number of classifiers in the system. For small numbers of classifiers—where small is relative to the number of potential messages—the average number of inputs should not change with additional classifiers. However, as the number of classifiers relative to potential messages increases, multiple classifiers sharing the same internal node result in greatly increased connectivity. Figure 4 indicates that pass-through inhibits connectivity.

This occurs because a higher number of internal nodes are created from pass-through for the different possible output messages, and they share the same number of connections. One unexpected result of pass-through is an increase in the internal to internal node connectivity (see Figure 5). With pass-through, the ratio of internal to external nodes in the system increases, and thus a random set of connections will have a higher rate of internal connections.

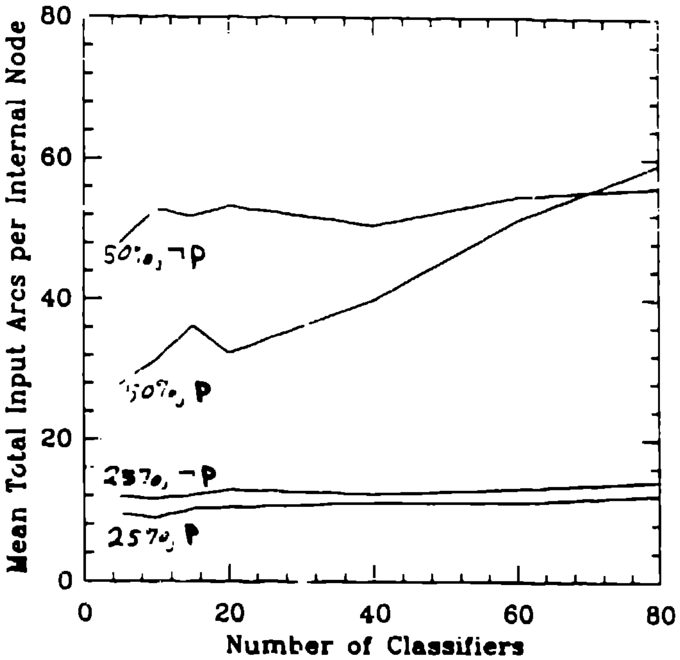

Figure 4. Mean Total Input Arcs per Internal Node

Another measure of internal connectivity is the property of self-sustaining activity (see Section 2). To test this property, all internal nodes were initialized to State 1 and the network was iterated until it reached a state cycle or activity died out (all nodes in State 0). Activity in sparsely connected networks will tend to die out, and highly connected networks will be self-sustaining. Once a self-sustaining cycle begins, it will continue in the absence of any exogenous inputs. Figure 6 shows the proportion of networks from different classifier system configurations that have self-sustaining components. Figure 6 indicate that a rapid transition occurs from systems without self-sustaining components to those that do have them. The average size of a self-sustaining cycle (the number of nodes active in the self-sustaining state cycle) is shown in Figure 7. This figure indicates a potential for rapid growth and saturation of internal networks.

3.3 Boolean Functions

The potential for messages to become frozen either on or off depends on the percentage of ones in the Boolean function associated with that message node (see Section 2.4). As the measure of "internal homogeneity" (Kauffman, 1984, p. 149) moves away from 50%, networks tend to exhibit large "frozen components" in which the states of nodes become

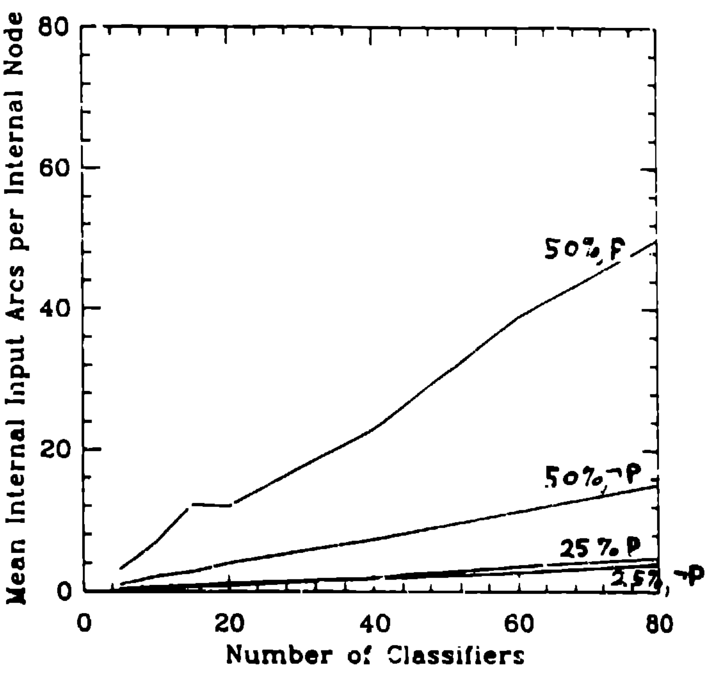

Figure 5. Mean Internal Input Arcs per Internal Node

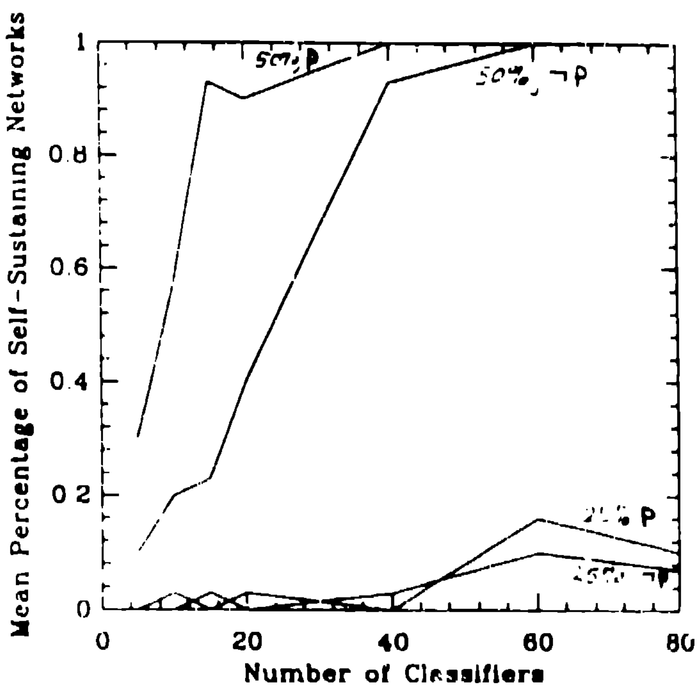

Figure 6. Percent of Networks with Self-Sustaining Activity

locked. For two-condition classifiers without negation that do not have pass through, the

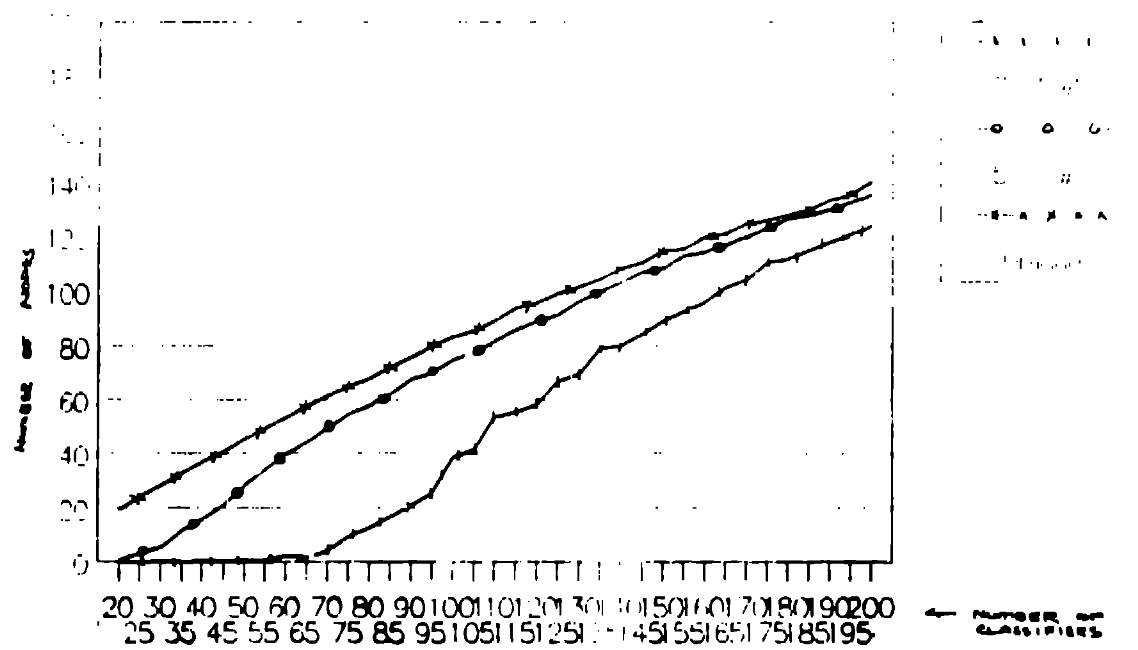

Figure 7. Mean Number of Nodes in the Self-Sustaining Cycle

percentage of one's in the truth table is given by

$$p_1 = \frac{(2^m - 1)(2^n - 1)}{2^{m+n}}, \tag{1}$$

where m and n denote the number of messages that could possibly match the first and second conditions respectively. This equation assumes that the set of matching messages for the two conditions is disjoint.[4] If the second condition is negated, then

$$p_2 = \frac{(2^m - 1)(1)}{2^{m+n}}. \tag{2}$$

Similar expressions can be written for Boolean functions that contain clauses for multiple classifiers (*i.e.*, when two or more classifiers share an output node), multiple classifiers that have identical conditions, *etc.*. More complicated expressions can be derived for classifiers that violate the assumption of disjointness among conditions.

Figure 8 graphs the proportion of ones using Equation (1) and assuming that $m = n$. Under these conditions a very narrow range of values exists that avoids frozen components. If the percentage of ones in a node's truth table is between 30-70% then the probability of that node freezing is low. Assuming that Equation (1) is a reasonable approximation, the number of #s in each condition that will prevent freezing is as follows (f, $*$ imply freeze,

[4] The logic of the equation is as follows: the first condition will not be true iff all of the m conditions are false (since it is the disjunction of possible matching messages) and therefore it will be true for $2^m - 1$ of its possible 2^m states. Similarly, the second condition will hold for $2^n - 1$ of its states. Thus they will both hold (and the full Boolean function will be true) for $(2^m - 1)(2^n - 1)$ out of the 2^{m+n} states.

not freeze respectively):

$$
\begin{array}{c}
 \\ 0 \\ 1 \\ 2 \\ \geq 3
\end{array}
\begin{array}{c}
\begin{array}{cccc} 0 & 1 & 2 & \geq 3 \end{array} \\
\begin{pmatrix}
f & * & * & * \\
* & * & * & f \\
* & * & f & f \\
* & f & f & f
\end{pmatrix}
\end{array}
$$

Thus, if the first condition has no #s the node will be active when the second condition has at least one #. If the first condition has 2 #s then freezing will occur when the second condition has more than 1 #. The actual distribution of #s in each condition can be controlled when the system is initialized or through biases in the learning operators. These results suggest that classifier systems are highly sensitive to the proportion of #s in the population and that the nature of this sensitivity should be studied carefully. Preliminary analysis indicates that if #s are chosen randomly with probability 0.25 then about 53% of 8-bit system nodes and 95% of 16-bit system nodes will be frozen. With a 0.50 probability almost all of the nodes will be frozen in either system.

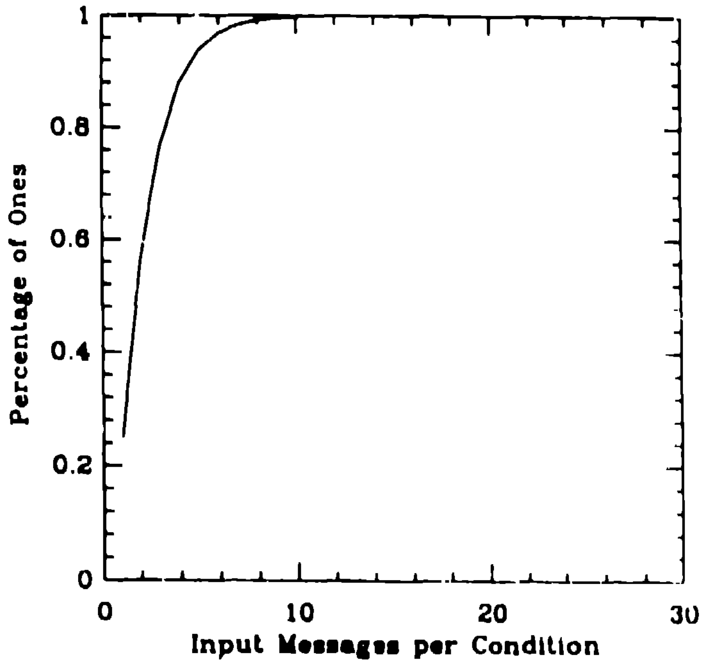

Figure 8. Percent of Ones in the Truth Table Using Equation (1) ($n = m$)

3.4 Learning

An important question is how the structural properties of networks differ for networks corresponding to randomly generated classifier systems and for classifier systems that have evolved under learning. We have obtained several pairs of classifier sets (before and after learning) and are in the process of comparing their structural properties. The final paper will present data and discuss the results of this comparison.

Another aspect of learning is how stable a classifier system is to the operations of the genetic algorithm. It is important, for example, that a random mutation or cross-over be capable of having a measurable effect on the overall system, but that it not completely disrupt all ongoing activity. In Boolean network terminology, it would be interesting to know whether one genetic algorithm operation were capable of unfreezing a set of frozen components (or freezing a set of unfrozen components). More generally, we are interested in the expected amount of perturbation caused by the application of the learning operators. This can be measured by comparing the dynamics of networks before and after the application of genetic operators.

3.5 Ongoing Work

The results reported here are partial, and we expect to supplement them with a variety of further experiments.

We are currently completing the studies of how basic classifier configurations relate to basic structural properties in Boolean networks. As part of this work it will be important to consider larger systems with 16 and 32 bits and work out the scaling relations between various-sized systems. There is a possibility of using the Connection Machine for the larger experiments. Once the basic structural properties are understood, it will be possible to carry out the actual dynamic analysis of the networks using RBNs.

The experiments to date have been conducted using the simple mapping between classifier systems and Boolean networks. Using the extended mapping, with restricted message lists and bidding, is an area of future investigation.

Another area of active investigation is determining the frequencies of 1's in Boolean functions corresponding to actual classifier systems. Once these frequencies are understood, it will be possible to assess the effect that frozen components are having on running classifier systems.

These basic studies are requisite to a careful investigation of how the Boolean networks differ before and after learning which we expect will be one of the most revealing aspects of this work.

4. Discussion

The methodology and specific results described in Sections 2 and 3 have many implications, some technical and some of a more general nature.

Of direct consequence, are the implications for the design of classifier systems. Constraints on parameter settings are already emerging from the results of Section 3 and continued work in this direction should lead to a theoretical treatment of this issue. By comparing initial classifier sets with final classifier sets (both those that have successfully solved a problem and those that have not), we also hope to analyze learning mechanisms from a new perspective. One open question is whether learning causes a classifier system to change its topology significantly or whether it is simply learning to use existing structures (cycles, attractors, *etc.*). Learning could be trying to overcome nearly impossible constraints imposed by the structural properties of the Boolean networks. In this case, changes in the underlying architecture might be appropriate to improve performance of the learning algorithms. Alternatively, careful study of classifier system dynamics could reveal unexploited structures and the learning could be tailored to exploit the inherent structures.

The mapping between classifier systems and Boolean networks is for analysis purposes only. Because representing every possible link in a system explicitly is highly inefficient, the mapping of Section 2 is not a likely alternative to classifier systems. However, Alternative network mappings exist, and they may be useful either for understanding other aspects of classifier systems or in their own right as models for cognitive activity.

The evidence that classifier system performance is highly sensitive to the proportion of #s suggests that an annealing (Kirkpatrick *et al.*, 1983) approach to this parameter might be useful. In initial stages of learning, higher connectivity would be advantageous as the system emphasizes exploration and identifies high-level default behaviors. At later periods, as the system refines its model of the environment with more specific rules, a lower proportion of #s would be more appropriate. This annealing scheme could be implemented by biasing the genetic operators.

The formation of default hierarchies is an important and controversial aspect of classifier systems. An area of future investigation is to study the formation of default hierarchies using the Boolean network framework. For example, the structural properties of networks built from classifiers using schemata with one defined bit (*e.g.*, 1#######) could be compared with those built with two bits defined, *etc*. Further, the dynamics of transitions from 1-bit to 2-bit schemata could be studied.

The distinction between internal and external nodes for Boolean networks raises the possibility that a system could learn to use its external nodes (*i.e.*, the environment) as a form of external memory. In this scenario, a classifier system might produce an external message that affected its environment in some predictable way (*e.g.*, causing the environment to produce some input message at a later time) and rely on that effect for later processing. Two examples illustrate how common this use of external memory is in the natural world, alarm clocks and pheromones. People do not have reliable internal clocks, but by setting an alarm clock before retiring (performing an action in an external environment) they can rely on its predictable behavior to awaken them at some time in the future and begin executing some internal process. Likewise, ants deposit various pheromones in their environment which they use later as trail markers to find the way back to their nest.

A dynamical systems approach to understanding learning phenomena and other cognitive activity may be very insightful. Until recently, there has been little theoretical work on learning systems. Most research has been of the "proof of principle" form in which a particular learning system is used to solve some example problem. Current theoretical research tends to emphasize *what* or how *efficiently* different systems can learn, but says little about *how* learning systems really work. The dynamical systems approach described here provides a way to answer these kinds of questions. Further, it can be applied to systems that cannot be easily understood analytically, either because they are too complicated or too messy. This paper applies the methodology to classifier systems, but similar mappings can be defined for neural networks and other learning schemes. Once a mapping has been defined, the techniques described here will readily apply. The unification of these seemingly disparate systems into a single class of models may have interesting consequences.

5. Conclusions

Classifier systems are quite complicated. A standard classifier system contains a computationally complete interior performance element, a bidding mechanism, two learning algorithms, and an input/output interface. Each of these parts interacts with the rest of the system in non-trivial ways. The complexity of the classifier system architecture is not surprising because it is intended to support a wide spectrum of activity. However, that complexity makes it difficult to understand precisely how each part of the system is contributing to overall performance.

The mapping from classifier systems to random Boolean networks provides a theoretical approach to the study of classifier system components and their interactions. The dynamic behavior of classifier systems is dominated by emergent properties which greatly influence their performance. Various configurations of classifier systems have different dynamic properties that vary in predictable ways. The general methodology derived here provides insights into the behavior of current classifier systems, and guidance for future extensions. More generally, the methodology is applicable to other learning systems such as neural networks, and it suggests several new avenues for exploring cognitive phenomena.

Bibliography

Derrida, B. and Stauffer, D. "Phase-transitions in two-dimensional Kauffman cellular automata." *Europhysics Letters* 2(1986):739–45.

Derrida, B. and Weisbuch, G. "Evolution of overlaps between configurations in random Boolean networks." *Journal Physique* 47(1986):1297–1303.

Forrest, S. *A Study of Parallelism in the Classifier System and Its Application to Classification in KL-ONE Semantic Networks*. Ph.D. Thesis, The University of Michigan, Ann Arbor, Michigan (1985).

Holland, J. H. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press:Ann Arbor, Michigan (1975).

Holland, J. H. "Searching Nonlinear Functions for High Values.' *Applied Mathematics and Computation*, forthcoming.

Kauffman, S. A. "Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets." *Journal of Theoretical Biology* 22(1969):437–67.

Kauffman, S. A. "Emergent Properties in Randomly Complex Automata." *Physica D* 10(1984):145–56.

Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P. "Optimization by Simulated Annealing." *Science* 220(May 13, 1983):671-80.

Riolo, R. Ph.D. Thesis, The University of Michigan, Ann Arbor, Michigan (1988).